IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of: § Group Art Unit: 2154

§

Gregory L. Slaughter, et al. § Examiner: Patel, Haresh N.

§

§ Atty. Dkt. No.: 5181-72200

§ P4987

Serial No. 09/693,672 §

§

§

Filed: · October 19, 2000 §

§

For: BRIDGING BEWTEEN A DATA §
REPRESENTATION LANGUAGE §
MESSAGE-BASED §
DISTRIBUTED COMPUTING §
ENVIRONTMENT AND OTHER §
ENVIRONMENTS §

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir/Madam:

Further to the Notice of Appeal filed April 13, 2006, Appellants present this Appeal Brief. Appellants respectfully request that the Board of Patent Appeals and Interferences consider this appeal.

09/22/2006 HVUONG1 00000017 501505 09693672
01 FC:1402 500.00 DA

I. **REAL PARTY IN INTEREST**

As evidenced by the assignment recorded at Reel/Frame 011250/0112, the subject application is owned by Sun Microsystems, Inc., a corporation organized and existing under and by virtue of the laws of the State of Delaware, and now having its principal place of business at 4150 Network Circle, Santa Clara, CA 95054.

II.   RELATED APPEALS AND INTERFERENCES

No other appeals, interferences or judicial proceedings are known which would be related to, directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

Claims 1-7, 12-24, 51-56, 61-73, 100-105, 110-117, 136 and 138 stand finally rejected. Claims 25-50, 74-99, 118-135, 137 and 139 have been canceled. Claims 8-11, 57-60 and 106 – 109 are objected to as being dependent on rejected base claims but allowable if rewritten in independent form. The rejection of claims 1-7, 12-24, 51-56, 61-73, 100-105, 110-117, 136 and 138 is being appealed. A copy of claims 1-24, 51-73, 100-117, 136 and 138 as currently pending is included in the Claims Appendix herein below.

IV. <u>STATUS OF AMENDMENTS</u>

No amendments to the claims have been submitted subsequent to the final rejection.

V.   SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 1 is directed to a method for bridging a first computing environment based on a message passing model to a second computing environment. The method of claim 1 includes a first entity in the first computing environment accessing a proxy service through messages in a data representation language. For example, foreign devices, clients, services, and transports may be bridged into a distributed computing environment using proxy services. In various embodiments, a proxy service may include a device, proxy, a client proxy, a service proxy or a transport proxy. As described in Appellants' specification, a device proxy may be a service that implements a device proxy protocol, such as SKP, Jini, UpnP, etc. A client proxy may be a service that implements a distributed computing environment proxy protocol on behalf of foreign client such as a browser, a service proxy may implement a distributed computing environment proxy protocol on behalf of a foreign service and a transport proxy may route data representation language messages between two different message transports. *See, e.g.*, FIG. 27, 30, 47, 48, 50 and 51; page 13, lines 3 –7; page 145, lines 1 – 11; page 149, line 27 – page 150, line 10; page 150, lines 15 – 25; page 150, line 27 – page 151, line 3; page 151, lines 5 – 16.

The method of claim 1 also includes the proxy service providing to the first entity an interface to a second entity in the second computing environment, wherein the proxy service appears to the first entity as the second entity and the first entity accessing the second entity in the second computing environment through the interface provided by the proxy service. For example, in some embodiments, a client may desire to access a service, but the client and service may not be able to communicate directly, such as because they may support different network or transport protocols. A proxy service may bridge the client and the service, such as by routing messages between the client and service. See, e.g. FIG. 27, 30, 47, 48, 50 and 51; page 50, lines 20 – 29; page 53, lines 23 – 29; page 91, line 29 – page 92, line 4; page 100, lines 22-28; page 145, lines 1 – 11; 147, lines 9 – 22; page 152, line 2 – page 154, line 21. The client may invoke a proxy service corresponding to the service. The proxy may be created dynamically. For

example, the client may download the proxy code and instantiate a new instance of the proxy in some embodiments. *See, e.g.*, FIG. 27, 30, 47, 48, 50 and 51; page 13, line 16 – page 14, line 17, page 50, lines 20 – 29; page 53, lines 23 – 29; page 80, lines 15-22; page 91, line 29 – page 92, line 4; page 100, lines 22-28; page 145, lines 1 – 11; 147, lines 9 – 22; page 152, line 2 – page 154, line 21.

Independent claim 51 is directed to a distributed computing system including a first device in a first computing environment based on a message passing module and a second device in a second computing environment not based upon the message passing model of the first environment. In some systems, there may be devices that do not support the message passing model of the rest of the distributed computing environment. However, these devices may provide services useful to client in the first computing environment. As discussed below, the distributed computing environment may include a mechanism to bridge such devices to clients so that the services offered on such devices may be accessed by the clients. *See, e.g.*, FIG. 27, 30, 47, 48, 50 and 51; page 13, lines 3 –7; page 141, lines 14 – 24; page 145, lines 1 – 11; page 149, line 27 – page 150, line 10; page 150, lines 15 – 25; page 150, line 27 – page 151, line 3; page 151, lines 5 – 16.

The system of claim 51 also includes a proxy service configured to provide an interface to the second device in the second computing environment to entities in the first computing environment, where the proxy service appears to the first entity as the second entity. For example, foreign devices, clients, services, and transports may be bridged into a distributed computing environment using proxy services. In various embodiments, a proxy service may include a device, proxy, a client proxy, a service proxy or a transport proxy. A device proxy may be a service that implements a device proxy protocol, such as SKP, Jini, UpnP, etc. A client proxy may be a service that implements a distributed computing environment proxy protocol on behalf of foreign client such as a browser, a service proxy may implement a distributed computing environment proxy protocol on behalf of a foreign service and a transport proxy may route data representation language messages between two different message transports. *See, e.g.*, FIG. 27, 30, 47, 48, 50 and 51; page 13, lines 3 –7; page 145, lines 1 – 11; page 149, line 27 – page 150, line 10;

page 150, lines 15 – 25; page 150, line 27 – page 151, line 3; page 151, lines 5 – 16.

The first device is configured to access the proxy service through messages in the data representation language and access the second device in the second computing environment through the interface provided by the proxy service. For example, a client may generate a message in a data representation language and the proxy may route the message to the service, potentially translating the message into a protocol supported by the service. A service may also publish a schema defining a set of messages that represent methods implemented by the service and the client may use the schema to generate messages, such as messages in a data representation language to access the service via the proxy service. *See, e.g.*, FIG. 27, 30, 47, 48, 50 and 51; page 13, line 16 – page 14, line 17, page 50, lines 20 – 29; page 53, lines 23 – 29; page 80, lines 15-22; page 91, line 29 – page 92, line 4; page 100, lines 22-28; page 145, lines 1 – 11; 147, lines 9 – 22; page 152, line 2 – page 154, line 21.

Independent claim 100 is directed to a carrier medium comprising program instructions that are computer executable to implement a first entity in the first computing environment accessing a proxy service through messages in a data representation language. For example, foreign devices, clients, services, and transports may be bridged into a distributed computing environment using proxy services. In various embodiments, a proxy service may include a device, proxy, a client proxy, a service proxy or a transport proxy. As described in Appellants' specification, a device proxy may be a service that implements a device proxy protocol, such as SKP, Jini, UpnP, etc. A client proxy may be a service that implements a distributed computing environment proxy protocol on behalf of foreign client such as a browser, a service proxy may implement a distributed computing environment proxy protocol on behalf of a foreign service and a transport proxy may route data representation language messages between two different message transports. *See, e.g.*, FIG. 27, 30, 47, 48, 50 and 51; page 13, lines 3 –7; page 145, lines 1 – 11; page 149, line 27 – page 150, line 10; page 150, lines 15 – 25; page 150, line 27 – page 151, line 3; page 151, lines 5 – 16.

The program instructions are also executable to implement the proxy service providing to the first entity an interface to a second entity in a second computing environment, where the proxy service appears to the first entity as the second entity and the first entity accessing the second entity in the second computing environment through the interface provided by the proxy service. For example, in some embodiments, a client may desire to access a service, but the client and service may not be able to communicate directly, such as because they may support different network or transport protocols. A proxy service may bridge the client and the service, such as by routing messages between the client and service. *See, e.g.*, FIG. 27, 30, 47, 48, 50 and 51; page 50, lines 20 – 29; page 53, lines 23 – 29; page 91, line 29 – page 92, line 4; page 100, lines 22-28; page 145, lines 1 – 11; 147, lines 9 – 22; page 152, line 2 – page 154, line 21. The client may invoke a proxy service corresponding to the service. The proxy may be created dynamically. For example, the client may download the proxy code and instantiate a new instance of the proxy in some embodiments. *See, e.g.*, FIG. 27, 30, 47, 48, 50 and 51; page 13, line 16 – page 14, line 17, page 50, lines 20 – 29; page 53, lines 23 – 29; page 80, lines 15-22; page 91, line 29 – page 92, line 4; page 100, lines 22-28; page 145, lines 1 – 11; 147, lines 9 – 22; page 152, line 2 – page 154, line 21.

Independent claim 136 is directed to a method for bridging a first computing environment based on a message passing model to a second computing environment. The method of claim 136 includes a first entity in the first computing environment accessing a proxy service through messages in a data representation language and the proxy service providing to the first entity an interface to a second entity in the second computing environment and the first entity accessing the second entity in the second computing environment through the interface provided by the proxy service. As described above regarding claim 1, foreign devices, clients, services, and transports may be bridged into a distributed computing environment using proxy services. In various embodiments, a proxy service may include a device, proxy, a client proxy, a service proxy or a transport proxy. As described in Appellants' specification, a device proxy may be a service that implements a device proxy protocol, such as SKP, Jini, UpnP, etc. A client proxy may be a service that implements a distributed computing environment

proxy protocol on behalf of foreign client such as a browser, a service proxy may implement a distributed computing environment proxy protocol on behalf of a foreign service and a transport proxy may route data representation language messages between two different message transports. *See, e.g.*, FIG. 27, 30, 47, 48, 50 and 51; page 13, lines 3 –7; page 145, lines 1 – 11; page 149, line 27 – page 150, line 10; page 150, lines 15 – 25; page 150, line 27 – page 151, line 3; page 151, lines 5 – 16.

In the method of claim 136, the proxy service providing to the first entity an interface to a second entity in the second computing environment includes providing an advertisement for the second entity, where the advertisement for the second entity includes access information for accessing the second entity in the second environment from the first environment. Ad advertisement may describe the content type or the capabilities of another entity, such as a service. In addition, an advertisement may describe the capabilities provided by a service. For example, an advertisement may specify a service's message schema an may include a URI address for locating the service. A client may discover and use such an advertisement to run the service, such as by sending (and receiving) messages to and from the service via the proxy service. *See, e.g.*, FIG. 9, 11B, 14, 15, 18, 22, 24, 27, 29, 31; page 28, lines 12 – 21; page 7 – 21; page 36, line 21 – page 37, line 3; page 45, lines 13 – 22; page 46, lines 7 – 17; page 51, line 20 – page 52, line 7; page 57, lines 1 – 9; page 64, lines 8 – 15; page 67, lines 8 –27.

Additionally, the advertisement includes information describing one or more computer programming language method calls to methods in the computer programming language provided by the second entity. For example, an advertisement may include schema information including definitions of one or more methods that may be invoked on a service. *See, e.g.*, FIG. 9, 11B, 14, 15, 18, 22, 24, 27, 29, 31; page 28, lines 12 – 21; page 29, lines 4 – 15; page 51, line 20 – page 52, line 7; page 56, lines 1 – 9; page 64, lines 8 – 15; page 67, lines 8 –27.

The method also includes constructing on the first entity a client method gate configured to provide an interface to the second entity by generating data representation

language message including information representing the method calls. For example, a method gate may support a form of remote method invocation. A method gate may provide a method interface between clients and services. For instance, a method gate may be generated from schema information, such as from a service advertisement. The schema information may include definitions of method interfaces and code may be generated as part of the method gate for interfacing to one or more methods. Each method invocation in the generated code may cause a message to be sent to the service containing marshaled method parameters. Thus, a method gate may provide a message interface to remotely invoke a service method and may be generated on a proxy or by a proxy service. In one embodiment, a method gate may be created in accordance with a message schema provided by a service, such as in a service advertisement. *See, e.g.*, FIG. 9, 11B, 14, 15, 18, 22, 24, 27, 29, 31; page 28, lines 12 – 21; page 29, lines 4 – 15; page 51, line 20 – page 52, line 7; page 56, lines 1 – 9; page 57, lines 11 – 29; page 59, lines 9 – 30; page 64, lines 8 – 15; page 67, lines 8 –27.

Independent claim 138 is directed to a method for bridging a first computing environment based upon a message passage model to a second computing environment. The method of claim 138 includes a first entity in the first computing environment accessing a proxy service through messages in a data representation language. As described above, a proxy service may include a device, proxy, a client proxy, a service proxy or a transport proxy. As described in Appellants' specification, a device proxy may be a service that implements a device proxy protocol. A client proxy may be a service that implements a distributed computing environment proxy protocol on behalf of foreign client such as a browser, a service proxy may implement a distributed computing environment proxy protocol on behalf of a foreign service and a transport proxy may route data representation language messages between two different message transports. *See, e.g.*, FIG. 27, 30, 47, 48, 50 and 51; page 13, lines 3 –7; page 145, lines 1 – 11; page 149, line 27 – page 150, line 10; page 150, lines 15 – 25; page 150, line 27 – page 151, line 3; page 151, lines 5 – 16.

The method also includes the proxy service providing to the first entity an

interface to a second entity in the second computing environment, where providing an interface includes sending to the first entity a schema defining one or more messages in the data representation language for accessing the second entity. For example, in some embodiments, a client may desire to access a service, but the client and service may not be able to communicate directly, such as because they may support different network or transport protocols. A proxy service may bridge the client and the service, such as by routing messages between the client and service. See, e.g. FIG. 27, 30, 47, 48, 50 and 51; page 50, lines 20 – 29; page 53, lines 23 – 29; page 91, line 29 – page 92, line 4; page 100, lines 22-28; page 145, lines 1 – 11; 147, lines 9 – 22; page 152, line 2 – page 154, line 21. A service may also publish a schema defining a set of messages that represent methods implemented by the service and the client may use the schema to generate messages, such as messages in a data representation language to access the service via the proxy service. See, e.g., FIG. 27, 30, 47, 48, 50 and 51; page 13, line 16 – page 14, line 17, page 50, lines 20 – 29; page 53, lines 23 – 29; page 80, lines 15-22; page 91, line 29 – page 92, line 4; page 100, lines 22-28; page 145, lines 1 – 11; 147, lines 9 – 22; page 152, line 2 – page 154, line 21.

The method of claim 138 further includes the first entity accessing the second entity in the second computing environment through the interface provided by the proxy service. The client may invoke a proxy service corresponding to the service. A service may also publish a schema defining a set of messages that represent methods implemented by the service and the client may use the schema to generate messages, such as messages in a data representation language to access the service via the proxy service. See, e.g., FIG. 27, 30, 47, 48, 50 and 51; page 13, line 16 – page 14, line 17, page 50, lines 20 – 29; page 53, lines 23 – 29; page 80, lines 15-22; page 91, line 29 – page 92, line 4; page 100, lines 22-28; page 145, lines 1 – 11; 147, lines 9 – 22; page 152, line 2 – page 154, line 21. For instance, a method gate may be generated from schema information, such as from a service advertisement. The schema information may include definitions of method interfaces and code may be generated as part of the method gate for interfacing to one or more methods. Each method invocation in the generated code may cause a message to be sent to the service containing marshaled method parameters. *See,*

e.g., FIG. 27, 30, 47, 48, 50 and 51; page 13, lines 3 –7; page 145, lines 1 – 11; page 149, line 27 – page 150, line 10; page 150, lines 15 – 25; page 150, line 27 – page 151, line 3; page 151, lines 5 – 16.

The summary above describes various examples and embodiments of the claimed subject matter; however, the claims are not necessarily limited to any of these examples and embodiments. The claims should be interpreted based on the wording of the respective claims.

VI.   GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1.   Claims 1-24, 51-73, 100-117, 136 and 138 stand finally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-45 of U.S. Patent No. 6,868,447.

2.   Claim 138 stands finally rejected under 35 U.S.C. § 102(e) as being anticipated by Tuatini (U.S. Patent Application Publication No. 2002/0032783).

3.   Claims 1-5, 19-21, 23, 24, 51-55, 68-70, 72, 73, 100-103, 113, 114, 116 and 117 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Tuatini in view of Mead et al. (U.S. Patent 6,061,728) (hereinafter "Mead").

4.   Claim 136 stands finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Tuatini in view of Cheng (U.S. Publication 2001/0032273), Machin et al. (U.S. Publication 2002/0032806) (hereinafter "Machin") and Beck et al. (U.S. Patent 6,604,140) (hereinafter "Beck").

5.   Claims 6, 7, 56, 104 and 105 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Tuatini, Mead and Cheng in view of Beck.

6.   Claims 12-18, 61-67, and 110-112 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Tuatini, Mead, Cheng and Beck in view of Machin.

7.   Claims 22, 71 and 115 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Tuatini in view of appellants' admitted prior art (AAPA).

VII. ARGUMENT

First Ground of Rejection

Claims 1-24, 51-73, 100-117, 136 and 138 stand finally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-45 of U.S. Patent No. 6,868,447. Appellants traverse this rejection for at least the following reasons.

Regarding the obviousness-type double patenting rejection, **the Examiner has failed to state a *prima facie* obviousness-type double patenting rejection**. According to MPEP 804.II.B.1, to state a proper obviousness-type double patenting rejection, the Examiner should list the differences between **each** rejected claim and the claims of the other patent/application, and for **each** difference the Examiner should give the reasons why a person of ordinary skill in the art would conclude that the invention defined in the claim at issue would have been an obvious variation of the invention defined in a claim of the other patent/application. Simply noting a few similarities between the claims does not satisfy the Examiner's burden to state valid reasons (supported by evidence of record) why a person of ordinary skill in the art would conclude that the invention defined in the claim at issue would have been an obvious variation of the invention defined in a claim of the other patent/application. Nor has the Examiner specifically addressed **each difference** of the claim of the present application compared to the claim of the other application.

Instead, the Examiner merely states, "[a]lthough the conflicting claims are not identical, they are not patentably distinct from each other because the patent teaches the limitations as disclosed such that the interpretation of a first entity accessing a second entity through messages in a data representation language is equivalent to a first client sending a first message in to a first service and the first service generating a set of results in response to the first message, wherein the set of results are expressed in a data representation language and using a space, advertisement, XML, and URI." The Examiner clearly has not met the requirements (as stated in MPEP 804.II.B.1) to establish

a *prima facie* obviousness-type double patenting rejection.

Additionally, the Examiner notes the claims of U.S. Patent No. 6,868,447 do not include the limitations in regard to bridging and proxy service as recited in the claims of the present application. The Examiner contends that these limitations were well known and would be obvious and refers to the Tuatini reference for support. Appellants traverse the Examiner's assertion that these limitations were well known in the context of Appellants' claimed invention. Additionally, the Examiner's reliance on Tuatini for the double patenting rejection (and the § 102 and § 103 rejections discussion below) is misplaced as 1) Tuatini fails to teach or suggest the particular limitations of Appellants' claims (as discussed below regarding individual claim rejections) and 2) the Examiner has not properly shown that Tuatini qualifies as prior art. As to the several other references cited by the Examiner, these references may indicate that bridging and proxy services were well known *in other contexts*, but fail to show that bridging and proxy services were well known in the context of the particular limitations of Appellants' claimed invention.

In the Advisory Action, the Examiner has incorrectly stated that Appellants' previous argument was bridging and proxy service is *not well known in the art*. The Examiner has misunderstood Appellants' argument. Appellants are arguing that a bridging and proxy service was not well known *in the context of Appellants' claimed invention*. Bridging and proxy services may have been well known in other contexts, but Appellants assert that bridging and proxy services were not well known in the context of Appellants' claimed invention which recites a *specific combination* of features.

For at least the reasons above, Appellants respectfully request removal of the double patenting rejection.

Second Ground of Rejection

Claim 138 stands finally rejected under 35 U.S.C. § 102(e) as being anticipated by Tuatini (U.S. Patent Application Publication No. 2002/0032783). Appellants traverse this rejection for at least the following reasons.

Regarding claim 138, the rejection is improper because (among other reasons) the Examiner has not shown that Tuatini qualifies as a prior art reference. The Examiner has the burden of proof to produce the factual basis for the rejection. *In re Warner*, 154 USPQ 173, 177 (C.C.P.A. 1967), *cert. denied*, 389 U.S. 1057 (1968). The Tuatini patent was filed on January 2, 2001, after Appellants' filing date of October 19, 2000. Tuatini does claim the benefit of provisional applications filed December 30, 1999. However, the December 30, 1999 filing date can only be used as Tuatini's 35 U.S.C. § 102(e) prior art date for the subject matter that is common to both the Tuatini patent and the provisional applications. Since it is common practice for a later filed utility application to include more or different subject matter than its earlier provisional application(s), it is unclear whether the material in Tuatini relied upon by the Examiner was actually present in Tuatini's provisional applications. In fact, from even a cursory review it is clear that Tuatini's published application differs greatly from its provisional applications.

Moreover, the Tuatini publication is not entitled to the December 30, 1999 date as a section 102(e) prior art date unless at least one claim of the Tuatini publication is supported (under 35 U.S.C. § 112) in the provisional application. Under 35 U.S.C. 119(e)(1), a published patent application is not entitled to its provisional application's filing date as a prior art date unless at least one claim of the published application is supported (per 35 U.S.C. § 112) in the provisional application. The rejection is improper unless the Examiner can show that Tuatini's published application has the necessary claim support in the provisional application to be entitled to the provisional application's filing date as its § 102(e) prior art date. *See also* M.P.E.P. § 2136.03(IV).

In response to Appellants' previous request that the Examiner particularly point out those portions of Tuatini's provisional application relied upon by the Examiner, as required by 37 CFR 1.104(c)(2), the Examiner refers to pages 4, 16, 78, 112, 236, 324, and 428 of Tuatini's provisional application 60/173,712. **However, as with the other pages cited by the Examiner from the 60/173,712 provisional application, none of these pages support the subject matter relied upon by the Examiner in the rejection of claim 138.** Three of these pages (78, 112, 136) are portions of a general, high level, overview and do not support, or have any relevance to, the portions of Tuatini's publication that the Examiner is relying upon in his rejection. For example, page 78 is merely a title page that mentions "GEPS E-Business Architecture Workout" and "Java Coding Standards". Page 112 is an illustration depicting various categories, such as auditing, security, event logging, etc. illustrated as puzzle pieces with an "application component" puzzle piece in the center. Page 236 lists various developmental steps, such as "define and develop partitioning, replication and referral policies", "develop load, maintenance and synchronization services", "develop framework LDAP query components", etc. Nothing in these three pages supports the specific subject matter of Tuatini relied upon by the Examiner in his rejections.

The Examiner also refers to pages 4, 16 and 428 of Tuatini's provisional application 60/173,712. Page 4 illustrates the logical connects between various components that collaborate with Tuatini's ControlServlet component. Page 16 describes various Java classes, such as the Application, BaseGenericServlet and BaseHttpServlet classes. Page 428 provides a discussion of Tuatini's messaging component and is the only page referred to by the Examiner that teaches something relied upon by the Examiner, namely Tuatini's messaging component. The pages referred to by the Examiner, whether considered individually or as a group, do not support hardly any of the subject matter of Tuatini relied upon by the Examiner in his rejection.

Page 324 of Tuatini's provisional application 60/173,712 illustrates a flowchart. However, the details of the flowchart are completely illegible. Thus, page 324 (as it is illegible) does support any of the subject matter relied upon by the Examiner in his

rejection. In response to Appellants' earlier argument that page 324 is illegible, the Examiner additionally cites pages 305 – 323 of Tuatini's provisional application. These pages first describe pseudo-code of various class definitions, such as orderquery and order, in Tuatini's system as well as various aspects of Tuatini's schema compiler, such as the serialize and deserialize functions, the overall processing of the parser component, and the code generator component. The pages also describe the processing of a service request that is passed and may return serialized messages. **However, as with the other pages of the 60/173,712 provisional application cited by the Examiner, none of these pages support the subject matter relied upon by the Examiner in the rejection of claim 138.**

Furthermore, the pages of the Tuatini provisional application cited by the Examiner do not support any of the claims of Tuatini's published application. For instance, the pages cited by the Examiner do not support claim 1 of the Tuatini application per the requirements of 35 U.S.C. § 112, first paragraph. The rejection is improper unless the Examiner can show that Tuatini's published application has the necessary claim support in the provisional application to be entitled to the provisional application's filing date as its § 102(e) prior art date. *See also* M.P.E.P. § 2136.03(IV). The Examiner has not met his burden of proof to show that Tuatini qualifies as prior art for this reason as well.

In further regard to claim 138, regardless of whether or not Tuatini qualifies as prior art, Tuatini does not disclose <u>the proxy service</u> providing to the first entity an interface to a second entity in the second computing environment, wherein providing an interface comprises <u>sending to the first entity a schema defining one or more messages</u> in the data representation language for accessing the second entity. The Examiner cites paragraphs [0166-0168] where Tuatini describes the use of an LDAP directory service. However, Tuatini does not teach sending a client application (which the Examiner equates to a first entity) a schema defining one or more messages in the data representation language for accessing the second entity. Instead, Tuatini describes how a LDAP directory may include a schema defining object classes of information that

can be stored in the directory entries. Tuatini does not mention anything about a schema *defining messages in a data representation language* for accessing the LDAP directory service, as would be required for the Examiner's line of reasoning to be correct. Tuatini also fails to mention sending such a schema to the client application. Furthermore, Tuatini teaches how a client accesses a LDAP directory by instantiating a directory manager object and uses methods of the directory manager object to retrieve other objects (both directory entry objects and adapter objects) for accessing particular directory entries. (Tuatini, paragraph [0167]). **Thus, Tuatini's system provides access objects for the entries of a LDAP directory service rather than sending a schema defining messages in a data representation language for accessing the directory service.**

In the Response to Arguments section of the Final Office Action and the Advisory Action, the Examiner responds to Appellants' arguments above (as presented previously) by asserting that certain language used in Appellants' argument "are not recited in the rejected claims." The Examiner has misunderstood Appellants' argument. For example, the Examiner contends that sending "a schema to the <u>client application</u>" (underlining by Examiner) is not recited in Appellants' claim. However, Appellants never argued that sending a schema to the client application was recited in claim 138. The Examiner, in the rejection of claim 138 equates Tuatini's client component (FIG. 41) with the first entity of Appellants' claim. Appellants' argument is that Tuatini does not teach sending a schema defining messages in a data representation language to (Tuatini's) client application. Thus, Appellants' use of the phrase "to the client application" illustrates that Tuatini fails, **under the Examiner's interpretation and line of reasoning,** to disclose the specific limitations of claim 138. Similarly, the use of the terms, "for accessing the directory service" and "sending the XML DTD to a client component" in Appellants' remarks and arguments illustrate that the Examiner's reliance on Tuatini is misplaced.

The Examiner also cites paragraphs [0122-0132] where Tuatini describes the use of a XML document type definition (DTD) to specify message parameters used to request service functions. However, the cited passage does not mention a proxy service sending the XML DTD to a client component, which would be required if the Examiner's

interpretation of Tuatini were correct. Instead, Tuatini describes that the XML DTD may be a part of a group of information for each shared service providing functionality to clients and that the information is "made available to others" (Tuatini, paragraph [0125]). The mere statement that an XML DTD is *made available* to others does not disclose the specific limitation of *a proxy service sending* a schema to a first entity, as recited in Appellants' claim 138. There are, in fact, many ways in which information may be "made available" to entities in a distributed computing environment, as is well known in the art. For example, Tuatini states that the XML DTDs may be stored separately from the access interface information (Tuatini, paragraph [0128]) and that Tuatini's messaging component may *retrieve* the XML DTD to verify that a message is properly formatted, thus implying that in Tuatini's system the XML DTDs are made available by storing them in a shared location.

In the Response to Arguments and the Advisory Action, the Examiner incorrectly characterizes Appellants' previous argument by stating, "it seems that appellant is implying that the cited references do not disclose 'proxy client' / 'accessing of a Jini Service proxy from clients', which the claims are not limited to." As stated above (and previously) Appellants argue, in part, that Tuatini does not disclose a proxy service providing to the first entity an interface to a second entity ..., wherein providing an interface comprises sending to the first entity a schema defining one or more messages in the data representation language for accessing the second entity. The Examiner cites Tuatini's teaches regarding the use of a XML DTD. However, as stated quite clearly above, Tuatini does not disclose a proxy service sending the XML DTD, which the Examiner equates to the schema of Appellants' claim, to Tuatini's client component, which the Examiner equates to the first entity of Appellants' claim. Instead, as noted above, Tuatini merely teaches that the XML DTD (schema) is *made available* and that Tuatini's messaging component may retrieve the XML DTD (schema). Merely making a XML DTD available such that it may be retrieved does not disclose the proxy service sending to the first entity a schema. Thus, Tuatini clearly fails to disclose the proxy service providing to the first entity an interface to a second entity in the second computing environment, wherein said providing an interface comprises sending to the

first entity a schema defining one or more messages in the data representation language for accessing the second entity.

As anticipation under 35 U.S.C. § 102 requires that the presence in a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim (*Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984)), Tuatini clearly does not anticipate appellants' claim 138. Thus, the rejection of claim 138 is not supported by the prior art and removal thereof is respectfully requested.

Third Ground of Rejection

Claims 1-5, 19-21, 23, 24, 51-55, 68-70, 72, 73, 100-103, 113, 114, 116 and 117 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Tuatini in view of Mead et al. (U.S. Patent 6,061,728) (hereinafter "Mead"). Appellants traverse this rejection for at least the following reasons. **The Examiner's rejection is improper because, as shown above, Tuatini has not been established to be prior art to the present application.** Additional arguments are presented below for different groups of claims under their respective subheadings.

Claims 1 - 5, 19, 21, 23, 24, 51 - 55, 68, 70, 72, 73, 100 - 103, 114, 116, and 117:

Further regarding claim 1, Tuatini in view of Mead fails to teach or suggest a proxy service providing to the first entity an interface to a second entity in the second computing environment, wherein the proxy service appears to the first entity as the second entity. The Examiner admits that Tuatini fails to teach a proxy service that provides an interface to a second entity and that appears to a first entity as the second entity and relies upon Mead, citing column 3, line 1 through column 4, line 24 of Mead. Mead teaches a system in which multiple proxy devices coordinate to communicate messages between local area networks via a wide area network using a transparent bridging system. Specifically, Mead teaches the use of a master proxy device that

mediates and selects which of the proxy devices should handle messages sent between a local area network and a wide area network.

The Examiner's assertion that Mead teaches the use of a proxy service that appears to a first entity as a second entity is erroneous. Mead's proxy devices do not appear as other entities to Mead's clients (nor to any other entity), even if combined with the teachings of Tuatini. Nowhere does Mead mention that his proxy devices appear as other entities to components of Mead's system. Instead, Mead's proxy devices route messages received from an end station between two local area networks via a wide area network. Each proxy device routes messages and translates them between an Ethernet protocol and a TCP/IP protocol (Mead, FIG. 3 and column 6, lines 28-60). Mead does not mention that a proxy device appears to the end station as the entity to which the end station is sending a message, as would be required if Mead were to teach a proxy service that appears to a first entity as a second entity, as the Examiner contends.

The Examiner is apparently relying upon the fact that Mead's system includes a transparent bridging mechanism. However, transparent bridging is well understood in the art and **does not include** a proxy service that provides an interface to a second entity and that *appears as the second entity to a first entity.* Mead's proxy devices are transparent because an entity on one local area network sending a message to another local area network via a wide area network is not aware that the proxy devices are performing the routing. Instead, as noted above, Mead's proxy devices only route network message frames from one network to another. The end stations in Mead's system, even if combined with Tuatini, are not aware of Mead's proxy devices at all and do not view the proxy devices as some other entity in the computing environment.

In response to Appellants' argument above (as presented previously) the Examiner, in the Response to Arguments of the Final Office Action and the Advisory Action, asserts, "Mead's teachings and disclosure are not limited to the applicant[s'] assertions" (underlining by Examiner). The Examiner also repeats the assertion that Mead discloses a proxy service that appears to the first entity as the second entity, citing

the same passage (column 3, line 1 – column 4, line 24) as cited in the rejection of claim 1. However, the Examiner fails to make any substantive rebuttal or additional argument regarding the fact that Mead's proxy devices function as a transparent bridging mechanism, and not as recited in claim 1. Nor does the Examiner substantively rebut Appellants' argument that transparent bridging systems do not include proxy services that appear as a second entity to a first entity. Instead, the Examiner refers to a computer dictionary definition of various terms, such as proxy, bridge, schema, etc. However, none of the definitions in the cited reference describe or mention anything about a proxy service that appears as a second entity to a first entity and thus fail to support the Examiner's rejection. Nor does the cited reference describe a transparent bridging mechanism.

Additionally, the Examiner's statements regarding Appellants' claims containing "broadly claimed subject matter" reading on the Examiner's interpretation is clearly incorrect. As noted above (and which the Examiner has failed to properly rebut) Appellants' clearly state that Tuatini in view of Mead fails to teach or suggest a proxy service providing to the first entity an interface to a second entity in the second computing environment, wherein the proxy service appears to the first entity as the second entity. Appellant has demonstrated that Mead's proxy devices, which the Examiner equates to the proxy service of Appellants' claim, do not teach or suggest a proxy service that appears to a first entity as a second entity, even if combined with Tuatini. The Examiner has never provided any interpretation that shows how Mead's proxy devices, which as admitted by the Examiner provide a *transparent* bridging service, can appear as a second entity to a first entity.

Additionally, the Examiner's proposed combination of Tuatini and Mead would not result in a system that includes a proxy service providing to a first entity an interface to a second entity where the proxy service appears to the first entity as the second entity. Instead, the Examiner's proposed combination of Tuatini and Mead would result only in allowing Tuatini's application framework, including the messaging component to also transparently route messages between local area networks via a wide area networks using

the multiple proxy devices of Mead. Since neither Tuatini nor Mead, whether considered single or in combination, teaches or suggests a proxy service that appears as another entity, no combination of Tuatini and Mead would include such a proxy service (that appears as another entity).

Moreover, Mead's proxy devices are at a completely different computing layer than Tuatini's messaging component, which the Examiner interprets as the proxy service of Appellants' claim. Tuatini's messaging component does not have anything to do with routing frames between a LAN and a WAN. Even if one where to modify the messaging component of Tuatini, which the Examiner interprets as a proxy service providing to first entity an interface to a second entity, the result would merely allow Tuatini's messaging component to route messages between a local area network and a wide area network and between an Ethernet protocol and a TCP/IP protocol. Nothing in such a combination would include or suggest that the messaging component would appear as another entity.

Therefore, for at least the reasons presented above, the rejection of claim 1 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks also apply to claims 51 and 100.

Claims 20, 69 and 113:

Regarding claim 20, **Tuatini in view of Mead fails to teach or suggest where the second environment is a non-message based environment**. The Examiner cites FIG. 41 and paragraphs 122 – 160 of Tuatini. However, the cited passages do not teach or suggest a second environment that is a non-message based environment. The Examiner refers to Tuatini's system including, "CORBA server or Web Server using other than message based languages". However the Examiner's interpretation of Tuatini as well as CORBA and Web Servers is clearly incorrect.

Firstly, Tuatini's invention is clearly directed towards message-based systems. For example, Tuatini teaches, "[w]hen an application component needs to access

functionality provided by remote shared services, the component uses a local *messaging service*" (italics added, Tuatini, Abstract). Additionally, the Examiner's cited passage describes various aspects of Tuatini's message passing system. Nothing is mentioned about a non-message based environment.

Secondly, the Examiner's reference to a CORBA server or Web Server is misplaced. As is well understood in the art, both CORBA servers and Web Servers rely on message-based environments. For example, Tuatini states, "the transport connector will be executed <u>to control the sending of the message</u> to the shared service function and to control the <u>receiving of any response messages</u>". Tuatini continues by stating, "the transport connector will contain specialized knowledge specific to the transport service (e.g., ... **CORBA**, ... **HTTP**, ... etc.) to be used to communicate with the shared service, such as how to establish a connection and <u>how to send and receive messages</u>" (emphasis added, Tuatini, paragraph [0134]). Additionally, Tuatini teaches, "the transport connector (or service adapter) then sends the sub-message to the shared service through a connector interface 4115 for that type of shared service 4180 (e.g., ... a CORBA component provided by a CORBA server, ... a Web application provided by a Web server, ...), with the sub-message sent in a manner appropriate to execute the function for that transport service" (emphasis added, Tuatini, paragraph [0137]). Thus, Tuatini clearly teaches that both CORBA and HTTP (e.g. used by Web servers) rely on message-based systems.

Mead, not relied upon by the Examiner in the rejection of claim 20, also fails to teach or suggest a second environment that a non-message based environment. Thus, Mead fails to overcome the above mentioned deficiencies of Tuatini regarding a second environment that a non-message based environment. Therefore, the Examiner's combination of Tuatini and Mead fails to teach or suggest the limitation of claim 20.

For at least the reasons above, the rejection of claim 20 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks also apply to claims 69 and 113.

Fourth Ground of Rejection

Claim 136 stands finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Tuatini in view of Cheng (U.S. Publication 2001/0032273), Machin et al. (U.S. Publication 2002/0032806) (hereinafter "Machin") and Beck et al. (U.S. Patent 6,604,140) (hereinafter "Beck"). Appellants traverse this rejection for at least the following reasons.

Regarding claim 136, Tuatini in view of Cheng, Machin and Beck fails to teach or suggest that a proxy service providing to the first entity an interface to a second entity in the second computing environment comprises <u>providing an advertisement for the second entity</u>, wherein the advertisement for the second entity <u>includes access information for accessing the second entity</u> in the second environment from the first environment and wherein the advertisement <u>includes information describing one or more computer programming language method calls</u> to methods in the computer programming language provided by the second entity.

The Examiner admits that Tuatini fails to teach this limitation of claim 136. The Examiner cites FIG. 3 and paragraphs 9-12 and 23-24 of Cheng. However, the cited portions of Cheng do not describe providing an advertisement *including access information and information describing computer programming method calls*. Cheng teaches the use of thin glue layers to bridge a non-IP network with the Internet. Cheng's thin glue layers translate between the IP protocol and the non-IP protocol and allow commands and responses to tunnel between applications in the Internet and the non-IP network (Cheng, paragraph [0011]).

The Examiner seems to be arguing that Cheng's teachings regarding a HAVi (a particular non-IP network) application using a HAVi API to access Internet services implies *providing an advertisement* including access information and describing method calls. However, Cheng does not mention providing any sort of *advertisement that*

includes access information or describing computer programming language method calls. Instead, Cheng only refers to the fact that the glue layers can translate between the two protocols. As noted above, the Examiner admits that Tuatini fails to provide an advertisement including access information and describing computer programming language method calls. Thus, Cheng, even if combined with Tuatini, fails to teach providing an advertisement including access information and describing computer programming language method calls. Furthermore, Machin and Beck fail to overcome the above noted deficiencies of both Tuatini and Cheng. Therefore, the combination of Tuatini, Cheng, Machin and Beck fails to teach or suggest a proxy service providing an advertisement including access information and describing method calls.

Moreover, claim 136 corresponds to the subject matter of claim 8 written in independent form. Claim 8 is objected to as being dependent from a rejected base claims, but the Examiner states it would be allowable if rewritten in independent form. Thus, claim 136 should also be allowable.

Thus, for at least the reasons above, the rejection of claim 136 is not supported by the cited art and removal thereof is respectfully requested.

Fifth Ground of Rejection

Claims 6, 7, 56, 104 and 105 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Tuatini, Mead and Cheng in view of Beck. Appellants traverse this rejection for at least the reasons presented above regarding their respective independent claims.

Sixth Ground of Rejection

Claims 12-18, 61-67, and 110-112 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Tuatini, Mead, Cheng and Beck in view of Machin et al. (U.S. Publication 2002/0032806) (hereinafter "Machin"). Appellants traverse this rejection for

at least the following reasons. Different groups of claims are addressed under their respective subheadings.

Claims 12 and 61:

Regarding claim 12, the Examiner states that claim 8, from which claim 12 depends would be allowable if rewritten in independent form. Thus, since claim 12 depends from claim 8, claim 12 should also be allowable if rewritten in independent form.

Similarly, regarding claim 61, the Examiner states that claim 57, from which claim 61 depends would be allowable if rewritten in independent form. Thus, since claim 61 depends from claim 57, claim 61 should also be allowable if rewritten in independent form.

Thus, claims 12 and 61 should also be allowable if written in independent form.

Claims 13 – 15, 62 – 64 and 110 - 111:

Appellants traverse the rejection of claims 13, 14, 15, 62, 63, 64, 110 and 111 for at least the reasons presented above regarding their respective, independent claims.

Claims 16, 18, 65, 67 and 112:

In regards to claim 16, the Examiner has failed to provide a proper rejection of claim 16. The Examiner rejects claim 16 as part of a rejection of claim 9-18, 59-67 and 107-112. However, the Examiner does not mention anything regarding the limitations of claim 16. Nor does the Examiner cite any portions of Tuatini, Mead, Cheng, Beck or Machine that teaches or suggest the limitations of claim 16. In short, the Examiner has completely ignored the limitations of claim 16, which is clearly improper.

Furthermore, the Examiner's combination of Tuatini, Mead, Cheng, Beck and Machin **fails to teach or suggest <u>providing an advertisement for the stored data</u> to the first entity, wherein the advertisement for the stored data includes access information for the stored data**. Instead, Tuatini teaches that result information is transmitted directly to the requesting client via messages. For instance, Tuatini teaches that a container receives the request messages and forwards them to container adapter and "receives response messages from the container adapter and forwards them to the client system" (Tuatini, paragraph [0065]). Similarly, Tuatini also states, "[a]fter the message has been sent to execute the function of the shared service, the transport connector (or service adapter) then waits to receive response messages (if any) from the function" and that "[f]or each response message received, the transport connector performs the same processing discussed (e.g., translation, or additional security measures) in order to send the response message back to the requesting client" (Parenthesis in original, Tuatini, paragraph, [0138]).

Thus, Tuatini clearly and repeated states that result information is send in message to the requesting client. Nowhere does Tuatini mention providing an advertisement for the stored data to the first entity, where the advertisement includes access information for the stored data. Moreover, as noted above, the Examiner has failed to address the limitations of claim 16 in his rejection.

Thus, for at least the reasons above, the rejection of claim 16 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks also apply to claim 65.

Claims 17 and 66:

In regards to claim 17, the Examiner has failed to provide a proper rejection. As with the rejection of claim 16, the Examiner rejects claim 17 as part of a rejection of claim 9-18, 59-67 and 107-112. However, the Examiner does not mention anything regarding the limitations of claim 17. Nor does the Examiner cite any portions of

Tuatini, Mead, Cheng, Beck or Machine that teaches or suggest the limitations of claim 17. In short, the Examiner has completely ignored the limitations of claim 17, which is clearly improper.

Furthermore, the Examiner's combination of Tuatini, Mead, Cheng, Beck and Machin **fails to teach or suggest the first entity <u>accessing the advertisement for the stored data</u> and the first entity <u>accessing the stored data in accordance with the access information for the stored data in the advertisement for the stored data</u>.** Instead, as noted above regarding claim 16, Tuatini teaches that result information is transmitted directly to the requesting client via messages. For instance, Tuatini teaches that a container receives the request messages and forwards them to container adapter and "receives response messages from the container adapter and forwards them to the client system" (Tuatini, paragraph [0065]). Similarly, Tuatini also states, "[a]fter the message has been sent to execute the function of the shared service, the transport connector (or service adapter) then waits to receive response messages (if any) from the function" and that "[f]or each response message received, the transport connector performs the same processing discussed (e.g., translation, or additional security measures) in order to send the response message back to the requesting client" (Parenthesis in original, Tuatini, paragraph, [0138]).

Thus, Tuatini clearly and repeated states that result information is send in message to the requesting client. Nowhere does Tuatini mention providing an advertisement for the stored data to the first entity, where the advertisement includes access information for the stored data. Moreover, as noted above, the Examiner has failed to address the limitations of claim 17 in his rejection.

Thus, for at least the reasons above, the rejection of claim 17 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks also apply to claim 66.

Seventh Ground of Rejection

Claims 22, 71 and 115 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Tuatini in view of appellants' admitted prior art (AAPA), page 2-6 of the specification. Appellants traverse this rejection for at least the following reasons.

Claims 22, 71 and 115:

Regarding claim 22, the Examiner states that it would have been obvious to combine the teachings of Tuatini and the AAPA because the AAPA's use of Jini environment would provide access to the Jini services. Appellants disagree with the Examiner's statement. The Examiner's statement is entirely conclusory. Appellants submit that such a broad conclusory statement does not provide a sufficient motivation to combine the teachings Tuatini and the AAPA. "The factual inquiry whether to combine references must be thorough and searching." *McGinley v. Franklin Sports, Inc.*, 60 USPQ2d 1001, 1008 (Fed. Cir. 2001). It must be based on objective evidence of record. "This precedent has been reinforced in myriad decisions, and cannot be dispensed with." *In re Lee*, 61 USPQ2d 1430, 1433 (Fed. Cir. 2002). "A showing of a suggestion, teaching, or motivation to combine the prior art references is an essential component of an obviousness holding." *Brown & Williamson Tobacco Corp. v. Philip Morris Inc.*, 56 USPQ2d 1456, 1459 (Fed. Cir. 2000).

The Federal Circuit has stated: "[o]ur case law makes clear that the best defense against the subtle but powerful attraction of a hindsight-based obviousness analysis is rigorous application of the requirement for a showing of the teaching or motivation to combine prior art references." The need for specificity pervades this authority. *See, e.g., In re Kotzab*, 217 F.3d 1365, 1371, 55 USPQ2d 1313, 1317 (Fed. Cir. 2000) ("particular findings must be made as to the reason the skilled artisan, with no knowledge of the claimed invention, would have selected these components for combination in the manner claimed" (emphasis added)); *In re Rouffet*, 149 F.3d 1350, 1359, 47 USPQ2d 1453, 1459 (Fed. Cir. 1998) ("even when the level of skill in the art is high, the [Examiner] must

identify specifically the principle, known to one of ordinary skill, that suggests the claimed combination. In other words, the [Examiner] must explain the reasons one of ordinary skill in the art would have been motivated to select the references and to combine them to render the claimed invention obvious."); *See also, In re Fritch*, 972 F.2d 1260, 1265, 23 USPQ2d 1780, 1783 (Fed. Cir. 1992).

The Examiner has failed to provide any proper motivation for modifying Tuatini in view of AAPA. Instead the Examiner has merely pointed to standard boilerplate text indicating that Tuatini's system may be modified, but that <u>does not provide any motivation for the specific modification suggested by the Examiner</u>. Similarly, nothing from AAPA provides any suggestion to modify the teaching of Tuatini to include the Jini environment.

Just because the Jini environment was known in the prior art, does not mean that one of ordinary skill in the art would have been motivated to modify the teachings of Tuatini with the Jini environment. The Examiner has provided no objective evidence of record to the contrary. Additionally, "[t]he mere fact that references can be combined or modified does not render the resultant combination obvious unless the prior art also suggests the desirability of the combination" (M.P.E.P. § 2143.01, paragraph 8). The Examiner has only shown that both Tuatini and the Jini environment were known in the art. However, the Examiner's stated motivation, namely, "to utilize Jini services of the Jini environment so that a client will be able to acess [sic] advertisement related information from the remote servers of the Jini network through proxy services" amounts to nothing more than a conclusory statement based in hindsight analysis of the present application.

In the Response to Arguments and the Advisory Action, the Examiner notes that the test for obviousness to combine references is not whether the features of a secondary references may be bodily incorporated into the structure of a primary references and is also not that the claimed invention must be expressly suggested in any one or all of the references. Rather the test is what the combined teachings of the references would have

suggested to those of ordinary skill in the art. However, Appellants are not arguing that one reference may not be bodily incorporated into the structure of the other. Instead, Appellants are arguing that no one would be motivated to modify the system of Tuatini to include the Jini environment in view both the Tuatini reference and appellants' AAPA. Furthermore, this statement by the Examiner does not change the fact that the Examiner has **failed to provide a proper motivation** to modify Tuatini in view of appellants' AAPA. As noted above, the Examiner has merely referred to features of the Jini environment, such as "[t]he Jini environment would provide access to Jini services", "Jini services would provide information to the client over the network" and "[t]he client would utilize the provided information". Thus, the Examiner's stated motivation amounts to nothing more than concluding that since both Tuatini's system and the Jini environment were known it would be obvious to combine them, which as noted above, is clearly improper.

In light of the above remarks, Appellants assert that the rejection of claim 22 is not supported any evidence of record. Withdrawal of the rejection is respectfully requested. Similar remarks as discussed above in regard to claim 22 apply to claims 71 and 115.

CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 1-7, 12-24, 51-56, 61-73, 100-105, 110-117, 136 and 138 was erroneous, and reversal thereof is respectfully requested.

The Commissioner is authorized to charge the appeal brief fee of $500.00 and any other fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5181-72200/RCK. This Appeal Brief is submitted with a return receipt postcard.

Respectfully submitted,

Robert C. Kowert
Reg. No. 39,255
Attorney for Appellants

Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
(512) 853-8850

Date: September 18, 2006

VIII. CLAIMS APPENDIX

The claims on appeal are as follows.

1.   A method for bridging a first computing environment based upon a message passing model to a second computing environment, comprising:

a first entity in the first computing environment accessing a proxy service through messages in a data representation language;

the proxy service providing to the first entity an interface to a second entity in the second computing environment, wherein the proxy service appears to the first entity as the second entity; and

the first entity accessing the second entity in the second computing environment through the interface provided by the proxy service.

2.   The method as recited in claim 1, wherein the interface provides a data representation language messaging channel between the proxy service and the first entity in the first computing environment, and wherein the interface further provides a communications channel between the proxy service and the second entity in the second computing environment.

3.   The method as recited in claim 1, wherein the first entity is a client in the first computing environment, wherein the second entity is a service accessible through the second computing environment, and wherein the interface provided by the proxy service enables the first entity to access resources provided by the second entity to clients in the second environment.

4.   The method as recited in claim 3, wherein the proxy service providing to the first entity the interface to the second entity in the second computing environment

comprises locating the second entity among a plurality of services accessible through the second computing environment.

5.   The method as recited in claim 4, wherein said locating the second entity among the plurality of services accessible through the second computing environment comprises determining that the second entity includes information identifying the entity as a service accessible by entities in the first environment through proxy service interfaces to the second computing environment.

6.   The method as recited in claim 3, wherein the proxy service providing to the first entity an interface to a second entity in the second computing environment comprises providing an advertisement for the second entity, wherein the advertisement for the second entity includes access information for accessing the second entity in the second environment from the first environment.

7.   The method as recited in claim 6, further comprising:

publishing the advertisement for the second entity on a space in the first computing environment; and

wherein the first entity accessing a second entity in the second computing environment through an interface comprises:

the first entity accessing the advertisement for the second entity from one or more advertisements published on the space; and

the first entity accessing the second entity in accordance with the access information in the advertisement for the second entity.

8.   The method as recited in claim 6, wherein the advertisement includes

information describing one or more computer programming language method calls to methods in the computer programming language provided by the second entity, the method further comprising constructing on the first entity a client method gate configured to provide an interface to the second entity by generating data representation language messages including information representing the method calls.

9.   The method as recited in claim 8, wherein the first entity accessing the second entity comprises:

the first entity generating a method call in the computer programming language;

the client method gate generating a data representation language message including information representing the method call;

the client method gate sending the data representation language message to a proxy method gate comprised on the proxy service;

the proxy method gate generating one or more objects in the computer programming language from the information representing the method call; and

the proxy service invoking a method on the second entity, wherein the one or more objects are passed to the method in said invoking.

10.   The method as recited in claim 9, further comprising:

the second entity executing the invoked method, wherein said executing generates results data;

the second entity sending the results data to the proxy service.

11.   The method as recited in claim 10, further comprising:

the proxy service generating a results advertisement for the results data;

the proxy service sending the results advertisement to the client method gate; and

the first entity generating a results method gate from the results advertisement
sent to the client method gate.

12.   The method as recited in claim 8, wherein the computer programming
language is Java.

13.   The method as recited in claim 1, further comprising:

the first entity sending a first message in the data representation language to the
proxy service, wherein the first message includes information for the
second entity;

converting the first message in the data representation language to a first
transmission compatible with the second computing environment and
receivable by the second entity in the second computing environment; and

sending the first transmission to the second entity in the second computing
environment.

14.   The method as recited in claim 1, further comprising:

the second entity sending a first transmission compatible with the second
computing environment to the proxy service, wherein the first
transmission includes information for the first entity;

converting the first transmission to a first message in the data representation language, wherein the first message includes the information from the first transmission; and

sending the first message to the first entity in the first computing environment.

15. The method as recited in claim 14, wherein the first entity is a client in the first computing environment, wherein the second entity is a service accessible through the second computing environment, and wherein the information in the first transmission is results data generated by the second entity in response to a request sent to the second entity by the first entity through the proxy service.

16. The method as recited in claim 1, further comprising:

the second entity sending a first transmission compatible with the second computing environment to the proxy service, wherein the first transmission includes data for the first entity;

storing the data received in the first transmission; and

providing an advertisement for the stored data to the first entity, wherein the advertisement for the stored data includes access information for the stored data.

17. The method as recited in claim 16, further comprising:

the first entity accessing the advertisement for the stored data; and

the first entity accessing the stored data in accordance with the access information for the stored data in the advertisement for the stored data.

18.   The method as recited in claim 16, wherein the first entity is a client in the first computing environment, wherein the second entity is a service accessible through the second computing environment, and wherein the data in the first transmission is results data generated by the second entity in response to a request sent to the second entity by the first entity through the proxy service.

19.   The method as recited in claim 1, wherein the second environment is a message-based environment using a different language for messages than the data representation language used for messages in the first environment.

20.   The method as recited in claim 1, wherein the second environment is a non-message based environment.

21.   The method as recited in claim 1, wherein communication among entities in the second environment uses remote method invocation (RMI).

22.   The method as recited in claim 21, wherein the second environment is a Jini environment.

23.   The method as recited in claim 1, wherein the second environment is an enterprise computing environment, wherein the second entity is an enterprise service accessible through the enterprise computing environment.

24.   The method as recited in claim 1, wherein the data representation language is eXtensible Markup Language (XML).

51.   A distributed computing system, comprising:

a first device in a first computing environment based upon a message passing model;

a second device in a second computing environment not based upon the message passing model of the first environment; and

a proxy service configured to provide an interface to the second device in the second computing environment to entities in the first environment, wherein the proxy service appears to the first entity as the second entity;

wherein the first device is configured to:

access the proxy service through messages in the data representation language; and

access the second device in the second computing environment through the interface provided by the proxy service.

52. The system as recited in claim 51, wherein, in said providing an interface, the proxy service is further configured to:

provide a data representation language messaging channel between the proxy service and the first device in the first computing environment; and

provide a communications channel between the proxy service and the second device in the second computing environment.

53. The system as recited in claim 51, wherein the first device is a client in the first computing environment, wherein the second device is a service accessible through the second computing environment, and wherein the proxy service is further configured to enable the first device to access resources provided by the second device to clients in the second environment.

54.   The system as recited in claim 53, further comprising:

a plurality of services accessible through the second computing environment;

wherein, in said providing an interface, the proxy service is further configured to locate the second device among the plurality of services accessible through the second computing environment.

55.   The system as recited in claim 54, wherein, in said locating the second device among the plurality of services accessible through the second computing environment, the proxy service is further configured to determine that the second device comprises information identifying the device as a service accessible by entities in the first environment through proxy service interfaces to the second computing environment.

56.   The system as recited in claim 53, wherein, in said providing an interface, the proxy service is further configured to:

provide an advertisement for the second device in the second computing environment, wherein the advertisement for the second device includes access information for accessing the second device in the second computing environment from the first environment; and

wherein, in said accessing the second device in the second computing environment through the interface, the first device is further configured to:

access the advertisement for the second device; and

access the second device in accordance with the access information in the advertisement for the second device.

57.   The system as recited in claim 56, wherein the advertisement includes

information describing one or more computer programming language method calls to methods in the computer programming language provided by the second device, wherein the first device is further configured to construct a client method gate configured to provide an interface to the second device by generating data representation language messages including information representing the method calls.

58.   The system as recited in claim 57,

wherein, in said accessing the second device, the first device is further configured to generate a method call in the computer programming language;

wherein the client method gate is configured to:

generate a data representation language message including information representing the method call; and

send the data representation language message to a proxy method gate comprised on the proxy service;

wherein the proxy method gate is configured to generate one or more objects in the computer programming language from the information representing the method call; and

wherein the proxy service is further configured to invoke a method on the second device, wherein the one or more objects are passed to the method in said invoking.

59.   The system as recited in claim 58, wherein the second device is further configured to:

execute the invoked method, wherein said executing generates results data; and

send the results data to the proxy service.

60.   The system as recited in claim 59,

wherein the proxy service is further configured to:

generate a results advertisement for the results data; and

send the results advertisement to the client method gate; and

wherein the first device is further configured to generate a results method gate from the results advertisement sent to the client method gate.

61.   The system as recited in claim 57, wherein the computer programming language is Java.

62.   The system as recited in claim 51,

wherein the first device is further configured to send a first message in the data representation language to the proxy service, wherein the first message includes information for the second device; and

wherein the proxy service is further configured to:

convert the first message in the data representation language to a first transmission compatible with the second computing environment and receivable by the second device in the second computing environment; and

send the first transmission to the second device in the second computing environment.

63.   The system as recited in claim 51,

wherein the second device is configured to send a first transmission compatible with the second computing environment to the proxy service, wherein the first transmission includes information for the first device;

wherein the proxy service is further configured to:

convert the first transmission to a first message in the data representation language, wherein the first message includes the information from the first transmission; and

send the first message to the first device in the first computing environment.

64.   The system as recited in claim 63, wherein the first device is a client in the first computing environment, wherein the second device is a service accessible through the second computing environment, and wherein the information in the first transmission is results data generated by the second device in response to a request sent to the second device by the first device through the proxy service.

65.   The system as recited in claim 51,

wherein the second device is configured to send a first transmission compatible with the second computing environment to the proxy service, wherein the first transmission includes data for the first device;

wherein the proxy service is further configured to:

store the data received in the first transmission; and

provide an advertisement for the stored data to the first device, wherein the advertisement for the stored data includes access information for the stored data.

66.   The system as recited in claim 65, wherein the first device is further configured to:

access the advertisement for the stored data; and

access the stored data in accordance with the access information for the stored data in the advertisement for the stored data.

67.   The system as recited in claim 65, wherein the first device is a client in the first computing environment, wherein the second device is a service accessible through the second computing environment, and wherein the data in the first transmission is results data generated by the second device in response to a request sent to the second device by the first device through the proxy service.

68.   The system as recited in claim 51, wherein the second environment is a message-based environment using a different language for messages than the data representation language used for messages in the first environment.

69.   The system as recited in claim 51, wherein the second environment is a non-message based environment.

70.   The system as recited in claim 51, wherein communication among entities in the second environment uses remote method invocation (RMI).

71.   The system as recited in claim 70, wherein the second environment is a Jini environment.

72.   The system as recited in claim 51, wherein the second environment is an enterprise computing environment, wherein the second device is an enterprise service accessible through the enterprise computing environment.

73.   The system as recited in claim 51, wherein the data representation language is eXtensible Markup Language (XML).

100.   A carrier medium comprising program instructions, wherein the program instructions are computer-executable to implement:

a first entity in the first computing environment accessing a proxy service through messages in a data representation language;

the proxy service providing to the first entity an interface to a second entity in the second computing environment, wherein the proxy service appears to the first entity as the second entity; and

the first entity accessing the second entity in the second computing environment through the interface provided by the proxy service.

101.   The carrier medium as recited in claim 100, wherein the first entity is a client in the first computing environment, wherein the second entity is a service accessible through the second computing environment, and wherein the interface provided by the proxy service enables the first entity to access resources provided by the second entity to clients in the second environment.

102.   The carrier medium as recited in claim 101, wherein, in said providing to the first entity the interface to the second entity in the second computing environment, the program instructions are further computer-executable to implement locating the second entity among a plurality of services accessible through the second computing environment.

103.   The carrier medium as recited in claim 102, wherein, in said locating the second entity among the plurality of services accessible through the second computing environment, the program instructions are further computer-executable to implement determining that the second entity includes information identifying the entity as a service accessible by entities in the first environment through proxy service interfaces to the second computing environment.

104.   The carrier medium as recited in 101, wherein, in providing to the first entity an interface to a second entity in the second computing environment, the program instructions are further computer-executable to implement providing an advertisement for the second entity, wherein the advertisement for the second entity includes access information for accessing the second entity in the second environment from the first environment.

105.   The carrier medium as recited in claim 104,

wherein the program instructions are further computer-executable to implement publishing the advertisement for the second entity on a space in the first computing environment; and

wherein, in accessing a second entity in the second computing environment through an interface, the program instructions are further computer-executable to implement:

the first entity accessing the advertisement for the second entity from one or more advertisements published on the space; and

the first entity accessing the second entity in accordance with the access information in the advertisement for the second entity.

106. The carrier medium as recited in claim 104,

wherein the advertisement includes information describing one or more computer programming language method calls to methods in the computer programming language provided by the second entity;

wherein the program instructions are further computer-executable to implement constructing on the first entity a client method gate configured to provide an interface to the second entity by generating data representation language messages including information representing the method calls.

107. The carrier medium as recited in claim 106, wherein, in the first entity accessing the second entity, the program instructions are further computer-executable to implement:

the first entity generating a method call in the computer programming language;

the client method gate generating a data representation language message including information representing the method call;

the client method gate sending the data representation language message to a proxy method gate comprised on the proxy service;

the proxy method gate generating one or more objects in the computer programming language from the information representing the method call;

the proxy service invoking a method on the second entity, wherein the one or more objects are passed to the method in said invoking;

the second entity executing the invoked method, wherein said executing generates results data; and

the second entity sending the results data to the proxy service.

108. The carrier medium as recited in claim 107, wherein the program instructions are further computer-executable to implement:

the proxy service generating a results advertisement for the results data;

the proxy service sending the results advertisement to the client method gate; and

the first entity generating a results method gate from the results advertisement sent to the client method gate.

109. The carrier medium as recited in claim 107, wherein the computer programming language is Java.

110. The carrier medium as recited in claim 100, the program instructions are further computer-executable to implement:

the first entity sending a first message in the data representation language to the proxy service, wherein the first message includes information for the second entity;

converting the first message in the data representation language to a first transmission compatible with the second computing environment and receivable by the second entity in the second computing environment; and

sending the first transmission to the second entity in the second computing environment.

111. The carrier medium as recited in claim 100, the program instructions are further computer-executable to implement:

the second entity sending a first transmission compatible with the second computing environment to the proxy service, wherein the first transmission includes information for the first entity;

converting the first transmission to a first message in the data representation language, wherein the first message includes the information from the first transmission; and

sending the first message to the first entity in the first computing environment;

wherein the information in the first transmission is results data generated by the second entity in response to a request sent to the second entity by the first entity through the proxy service.

112. The carrier medium as recited in claim 100, wherein the program instructions are further computer-executable to implement:

the second entity sending a first transmission compatible with the second computing environment to the proxy service, wherein the first transmission includes data for the first entity;

storing the data received in the first transmission; and

providing an advertisement for the stored data to the first entity, wherein the advertisement for the stored data includes access information for the stored data;

wherein the data in the first transmission is results data generated by the second entity in response to a request sent to the second entity by the first entity through the proxy service.

113. The carrier medium as recited in claim 100, wherein the second environment is a non-message based environment.

114. The carrier medium as recited in claim 100, wherein communication among entities in the second environment uses remote method invocation (RMI).

115. The carrier medium as recited in claim 114, wherein the second environment is a Jini environment.

116. The carrier medium as recited in claim 100, wherein the second environment is an enterprise computing environment, wherein the second entity is an ·enterprise service accessible through the enterprise computing environment.

117. The carrier medium as recited in claim 100, wherein the data representation language is eXtensible Markup Language (XML).

136. A method for bridging a first computing environment based upon a message passing model to a second computing environment, comprising:

a first entity in the first computing environment accessing a proxy service through messages in a data representation language;

the proxy service providing to the first entity an interface to a second entity in the second computing environment;

the first entity accessing the second entity in the second computing environment through the interface provided by the proxy service;

wherein the proxy service providing to the first entity an interface to a second entity in the second computing environment comprises providing an advertisement for the second entity, wherein the advertisement for the second entity includes access information for accessing the second entity in the second environment from the first environment; and

wherein the advertisement includes information describing one or more computer programming language method calls to methods in the computer programming language provided by the second entity, the method further comprising constructing on the first entity a client method gate configured to provide an interface to the second entity by generating data representation language messages including information representing the method calls.

138. A method for bridging a first computing environment based upon a message passing model to a second computing environment, comprising:

a first entity in the first computing environment accessing a proxy service through messages in a data representation language;

the proxy service providing to the first entity an interface to a second entity in the second computing environment, wherein said providing an interface comprises sending to the first entity a schema defining one or more messages in the data representation language for accessing the second entity; and

the first entity accessing the second entity in the second computing environment through the interface provided by the proxy service.

IX.   <u>EVIDENCE APPENDIX</u>

No evidence submitted under 37 CFR §§ 1.130, 1.131 or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

X.   RELATED PROCEEDINGS APPENDIX

There are no related proceedings.